Package ‘alphaOutlier’

October 12, 2022

Type Package

Title Obtain Alpha-Outlier Regions for Well-Known Probability Distributions

Version 1.2.0

Date 2016-09-09

Author Andre Rehage, Sonja Kuhnt

Maintainer Andre Rehage <rehage@statistik.tu-dortmund.de>

Description Given the parameters of a distribution, the package uses the concept of alpha-outliers by Davies and Gather (1993) to flag outliers in a data set. See Davies, L.; Gather, U. (1993): The identification of multiple outliers, JASA, 88 423, 782-792, <doi:10.1080/01621459.1993.10476339> for details.

License GPL-3

Depends Rsolnp, nleqslv, quantreg, graphics

Imports stats

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-09 18:05:11

R topics documented:

`alphaOutlier-package` *Obtain α-outlier regions for well-known probability distributions*

Description

Given the parameters of a distribution, the package uses the concept of α-outliers by Davies and Gather (1993) to flag outliers in a data set.

Details

The structure of the package is as follows: `aout.[Distribution]` is the name of the function which returns the α-outlier region of a random variable following `[Distribution]`. The names of the distributions are abbreviated as in the `d`, `p`, `q`, `r` functions. Use pre-specified or robustly estimated parameters from your data to obtain reasonable results. The sample size should be taken into account when choosing `alpha`, for example Gather et al. (2003) propose $\alpha_N = 1-(1-\alpha)^{1/N}$.

Author(s)

A. Rehage, S. Kuhnt

References

Davies, L.; Gather, U. (1993) The identification of multiple outliers, *Journal of the American Statistical Association*, **88** 423, 782-792.

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

See Also

`nleqslv`, `solnp`, `rq.fit.fnc`

Examples

```
iris.setosa <- iris[1:51, 4]
aout.norm(data = iris.setosa, param = c(mean(iris.setosa), sd(iris.setosa)), alpha = 0.01)
aout.pois(data = warpbreaks[,1], param = mean(warpbreaks[,1]), alpha = 0.01,
          hide.outliers = TRUE)
```

`aout.binom` *Find α-outliers in Binomial data*

Description

Given the parameters of a Binomial distribution, `aout.binom` identifies α-outliers in a given data set.

Usage

```
aout.binom(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the Binomial distribution, N and p.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

See Also

`dbinom`

Examples

```
data(uis)
medbeck <- median(uis$BECK)
aout.binom(data = uis$BECK, param = c(54, medbeck/54), alpha = 0.001)
```

`aout.cg` *Find α-outliers in conditional Gaussian data*

Description

Given the parameters of a conditional Gaussian distribution, `aout.cg` identifies α-outliers in a given data set.

Usage

```
aout.cg(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a matrix. First column: Class of the value, coded with an integer between 1 and d, where d is the number of classes. Second column: The value as a realization of a univariate normal with parameters μ and σ. The data set to be examined.
`param`	a list with three elements: `p`: d-dimensional vector of probabilities of the classes. `mu`: d-dimensional vector of univariate mean values of each class. `sigma`: d-dimensional vector of univariate standard errors of each class
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If `hide.outliers` is set to `TRUE`, a data frame of the outlier-free data.

Author(s)

A. Rehage

References

Edwards, D. (2000) *Introduction to Graphical Modelling*. 2nd edition, Springer, New York.

Kuhnt, S.; Rehage, A. (2013) The concept of α-outliers in structured data situations. In C. Becker, R. Fried, S. Kuhnt (Eds.): *Robustness and Complex Data Structures. Festschrift in Honour of Ursula Gather.* Berlin: Springer, 91-108.

Examples

```
# Rats' weights data example taken from Edwards (2000)
ratweight <- cbind(Drug = c(1, 1, 2, 3, 1, 1, 2, 3, 1, 2, 3, 3, 1, 2, 2, 3, 1,
                            2, 2, 3, 1, 2, 3, 3),
                   Week1 = c(5, 7, 9, 14, 7, 8, 7, 14, 9, 7, 21, 12, 5, 7, 6,
                             17, 6, 10, 6, 14, 9, 8, 16, 10))
```

```
aout.cg(ratweight,
        list(p = c(1/3, 1/3, 1/3), mu = c(7, 7, 14), sigma = c(1.6, 1.4, 3.3)))
```

---

aout.chisq *Find α-outliers in χˆ2 data*

Description

Given the parameters of a χ^2 distribution, aout.chisq identifies α-outliers in a given data set.

Usage

```
aout.chisq(data, param, alpha = 0.1, hide.outliers = FALSE, ncp = 0, lower = auto.l,
           upper = auto.u, method.in = "Newton", global.in = "gline",
           control.in = list(sigma = 0.1, maxit = 1000, xtol = 1e-12,
                             ftol = 1e-12, btol = 1e-04))
```

Arguments

data	a vector. The data set to be examined.
param	an atomic vector. Contains the degrees of freedom of the χ^2 distribution.
alpha	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
hide.outliers	boolean. Returns the outlier-free data if set to TRUE. Defaults to FALSE.
ncp	an atomic vector. Determines the non-centrality parameter of the χ^2 distribution. Defaults to 0.
lower	an atomic vector. First element of x from nleqslv.
upper	an atomic vector. Second element of x from nleqslv.
method.in	See method in nleqslv.
global.in	See global in nleqslv.
control.in	See control in nleqslv.

Details

The α-outlier region of a χ^2 distribution is generally not available in closed form or via the tails, such that a non-linear equation system has to be solved.

Value

Data frame of the input data and an index named is.outlier that flags the outliers with TRUE. If hide.outliers is set to TRUE, a simple vector of the outlier-free data.

Author(s)

A. Rehage

See Also

dchisq

Examples

```
aout.chisq(chisq.test(occupationalStatus)$statistic, 49)
```

---

aout.conttab *Find α-outliers in two-way contingency tables*

Description

This is a wrapper function for aout.pois. We assume that each entry of a contingency table can be seen as a realization of a Poisson random variable. The parameter λ of each cell can either be set by the user or estimated. Given the parameters, aout.conttab identifies α-outliers in a given contingency table.

Usage

```
aout.conttab(data, param, alpha = 0.1, hide.outliers = FALSE, show.estimates = FALSE)
```

Arguments

data	a matrix or data.frame. The contingency table to be examined.
param	a character string from c("ML", "L1", "MP") or a vector containing the parameters of each cell of the Poisson distribution: λ. "ML" yields the maximum likelihood estimate from the log-linear Poisson model using a suitable design matrix. "L1" yields the L1-estimate from rq.fit.fnc. "MP" yields the Median Polish estimate. If the parameter vector is given by the user, it is necessary that the contingency table was filled byrow = FALSE.
alpha	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
hide.outliers	boolean. Returns the outlier-free data if set to TRUE. Defaults to FALSE.
show.estimates	boolean. Returns $\hat{\lambda}$ for each cell if set to TRUE. Defaults to FALSE.

Value

Data frame of the vectorized input data and, if desired, an index named is.outlier that flags the outliers with TRUE and a vector named param containing the estimated lambdas.

Author(s)

A. Rehage

References

Kuhnt, S. (2000) Ausreisseridentifikation im Loglinearen Poissonmodell fuer Kontingenztafeln unter Einbeziehung robuster Schaetzer. Ph.D. Thesis. Universitaet Dortmund, Dortmund. Fachbereich Statistik.

Kuhnt, S.; Rapallo, F.; Rehage, A. (2014) Outlier detection in contingency tables based on minimal patterns. *Statistics and Computing* 24 (3), 481-491.

See Also

`rq.fit.fnc`, `aout.pois`

Examples

```
aout.conttab(data = HairEyeColor[,,1], param = "L1", alpha = 0.01, show.estimates = TRUE)
aout.conttab(data = HairEyeColor[,,1], param = "ML", alpha = 0.01, show.estimates = TRUE)
```

---

aout.exp — *Find α-outliers in exponentially distributed data*

Description

Given the parameters of an exponential distribution, `aout.exp` identifies α-outliers in a given data set.

Usage

```
aout.exp(data, param, alpha = 0.1, hide.outliers = FALSE, theta = 0)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	an atomic vector. Contains the parameter of the exponential distribution.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.
`theta`	an atomic vector. Determines the lower bound of the support of the exponential distribution. Defaults to 0.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

See Also

dexp

Examples

```
aout.exp(attenu[,5], median(attenu[,5]), alpha = 0.05)
```

---

aout.gandh — *Find α-outliers in data from the family of g-and-h distributions*

Description

Given the parameters of a g-and-h distribution, `aout.gandh` identifies α-outliers in a given data set.

Usage

```
aout.gandh(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the g-and-h distribution: median, scale, g, h.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Details

The concept of α-outliers is based on the p.d.f. of the random variable. Since for g-and-h distributions this does not exist in closed form, the computation of the outlier region is based on an optimization of the quantile function with side conditions.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Note

Makes use of `solnp`.

Author(s)

A. Rehage

References

Xu, Y.; Iglewicz, B.; Chervoneva, I. (2014) Robust estimation of the parameters of g-and-h distributions, with applications to outlier detection. *Computational Statistics and Data Analysis* 75, 66-80.

Examples

```
durations <- faithful$eruptions
aout.gandh(durations, c(4.25, 1.14, 0.05, 0.05), alpha = 0.1)
```

`aout.hyper` *Find α-outliers in hypergeometric data*

Description

Given the parameters of a hypergeometric distribution, `aout.hyper` identifies α-outliers in a given data set.

Usage

```
aout.hyper(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the hypergeometric distribution: m, n, k.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

See Also

`Hypergeometric`

Examples

```
set.seed(1)
lotto6aus49 <- rhyper(100, 6, 43, 6)
aout.hyper(lotto6aus49, c(6, 43, 6), 0.1)
```

---

aout.kernel — *Find α-outliers in arbitrary univariate data using kernel density estimation*

Description

Given the arguments of the `density`, `aout.kernel` identifies α-outliers in a given data set.

Usage

```
aout.kernel(data, alpha, plot = TRUE, plottitle = "", kernel = "gaussian",
nkernel = 1024, kern.bw = "SJ", kern.adj = 1,
xlim = NA, ylim = NA, outints = FALSE, w = NA, ...)
```

Arguments

`data`	a vector. The data set to be examined.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain.
`plot`	boolean. If `TRUE`, a plot of the data and estimated density with shaded outlier region is printed.
`plottitle`	character string. Title of the plot.
`kernel`	See `kernel` in `density`.
`nkernel`	See `n` in `density`.
`kern.bw`	See `bw` in `density`.
`kern.adj`	See `adjust` in `density`.
`xlim`	a vector. Specify if you want to change the x-limits of the plot.
`ylim`	a vector. Specify if you want to change the y-limits of the plot.
`outints`	boolean. If `TRUE`, then the bounds of the inlier-regions and the chosen bandwidth are shown.
`w`	a vector. See `weights` in `density`.
`...`	Further arguments for `density` and `plot`.

Value

If outints = TRUE, a list of

Results
: A data frame containing one row for each observation. The observations are labelled whether they are outlying, the value of the estimated density at the observation is shown and the bound of the outlier identifier.

Bounds.of.Inlier.Regions
: The bounds of the inlier region(s).

KDE.Chosen.Bandwidth
: The bandwidth that was chosen by density.

Author(s)

A. Rehage

Examples

```
set.seed(23)
tempx <- rnorm(1000, 0, 1)
tempx[1] <- -2.5
aout.kernel(tempx[1:10], alpha = 0.1, kern.adj = 1, xlim = c(-3,3), outints = TRUE)
# not run:
# aout.kernel(tempx[1:200], alpha = 0.1, kern.adj = 1, xlim = c(-3,3))
```

---

aout.laplace *Find α-outliers in Laplace / double exponential data*

Description

Given the parameters of a Laplace distribution, aout.laplace identifies α-outliers in a given data set.

Usage

```
aout.laplace(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

data	a vector. The data set to be examined.
param	a vector. Contains the parameters of the Laplace distribution: μ, σ.
alpha	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
hide.outliers	boolean. Returns the outlier-free data if set to TRUE. Defaults to FALSE.

Value

Data frame of the input data and an index named is.outlier that flags the outliers with TRUE. If hide.outliers is set to TRUE, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Dumonceaux, R.; Antle, C. E. (1973) Discrimination between the log-normal and the Weibull distributions. *Technometrics*, 15 (4), 923-926.

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

Examples

```
# Using the flood data from Dumonceaux and Antle (1973):
temp <- c(0.265, 0.269, 0.297, 0.315, 0.3225, 0.338, 0.379, 0.380, 0.392, 0.402,
         0.412, 0.416, 0.418, 0.423, 0.449, 0.484, 0.494, 0.613, 0.654, 0.74)
aout.laplace(temp, c(median(temp), median(abs(temp - median(temp)))), 0.05)
```

---

aout.logis — *Find α-outliers in logistic data*

Description

Given the parameters of a logistic distribution, `aout.logis` identifies α-outliers in a given data set.

Usage

```
aout.logis(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the logistic distribution: μ, σ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Balakrishnan, N. (1992) Maximum likelihood estimation based on complete and type II censored samples. In N. Balakrishnan (Ed.): *Handbook of the Logistic Distribution*. Dekker, New York, 49-78.

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

See Also

dlogis

Examples

```
# Data example from Balakrishnan (1967)
lifetime <- c(785, 855, 905, 918, 919, 920, 929, 936, 948, 950)
aout.logis(lifetime, c(949.9, 63.44))
```

---

aout.mvnorm — *Find α-outliers in multivariate normal data*

Description

Given the parameters of a multivariate normal distribution, `aout.mvnorm` identifies α-outliers in a given data set.

Usage

```
aout.mvnorm(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a data.frame or matrix. The data set to be examined.
`param`	a list. Contains the parameters of the normal distribution: the mean vector μ and the covariance matrix σ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a data frame of the outlier-free data.

Author(s)

A. Rehage

References

Kuhnt, S.; Rehage, A. (2013) The concept of α-outliers in structured data situations. In C. Becker, R. Fried, S. Kuhnt (Eds.): *Robustness and Complex Data Structures. Festschrift in Honour of Ursula Gather.* Berlin: Springer, 91-108.

See Also

dnorm

Examples

```
temp <- iris[1:51,-5]
temp.xq <- apply(FUN = median, MARGIN = 2, temp)
aout.mvnorm(as.matrix(temp), param = list(temp.xq, cov(temp)), alpha = 0.001)
```

---

aout.nbinom *Find α-outliers in negative Binomial data*

Description

Given the parameters of a negative Binomial distribution, `aout.nbinom` identifies α-outliers in a given data set.

Usage

```
aout.nbinom(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the negative Binomial distribution: N, p.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

See Also

dnbinom, daysabs

Examples

```
data(daysabs)
aout.nbinom(daysabs, c(8, 0.6), 0.05)
```

---

aout.norm — *Find α-outliers in normal data*

Description

Given the parameters of a normal distribution, `aout.norm` identifies α-outliers in a given data set.

Usage

```
aout.norm(data, param = c(0, 1), alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the normal distribution: μ, σ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If `hide.outliers` is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

See Also

`dnorm`

Examples

```
iris.setosa <- iris[1:51, 4]
# implosion breakdown point:
aout.norm(data = iris.setosa, param = c(median(iris.setosa), mad(iris.setosa)),
          alpha = 0.01)
# better:
aout.norm(data = iris.setosa, param = c(median(iris.setosa), sd(iris.setosa)),
          alpha = 0.01)
```

`aout.pareto` *Find α-outliers in Pareto data*

Description

Given the parameters of a Pareto distribution, `aout.pareto` identifies α-outliers in a given data set.

Usage

```
aout.pareto(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the Pareto distribution: λ, θ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Details

We use the Pareto distribution with Lebesgue-density $f(x) = \frac{\lambda\theta^\lambda}{x^{\lambda+1}}$.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Gather, U.; Kuhnt, S.; Pawlitschko, J. (2003) Concepts of outlyingness for various data structures. In J. C. Misra (Ed.): *Industrial Mathematics and Statistics*. New Delhi: Narosa Publishing House, 545-585.

See Also

citiesData

Examples

```
data(citiesData)
aout.pareto(citiesData[[1]], c(1.31, 14815), alpha = 0.01)
```

---

aout.pois *Find α-outliers in Poisson count data*

Description

Given the parameters of a Poisson distribution, `aout.pois` identifies α-outliers in a given data set.

Usage

```
aout.pois(data, param, alpha = 0.1, hide.outliers = FALSE)
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameter of the Poisson distribution: λ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

See Also

dpois

Examples

```
aout.pois(data = c(discoveries), param = median(discoveries), alpha = 0.01)
```

`aout.weibull` *Find α-outliers in Weibull data*

Description

Given the parameters of a Weibull distribution, `aout.weibull` identifies α-outliers in a given data set.

Usage

```
aout.weibull(data, param, alpha = 0.1, hide.outliers = FALSE, lower = auto.l,
             upper = auto.u, method.in = "Broyden", global.in = "qline",
             control.in = list(sigma = 0.1, maxit = 1000, xtol = 1e-12,
                               ftol = 1e-12, btol = 1e-04))
```

Arguments

`data`	a vector. The data set to be examined.
`param`	a vector. Contains the parameters of the Weibull distribution: β, λ.
`alpha`	an atomic vector. Determines the maximum amount of probability mass the outlier region may contain. Defaults to 0.1.
`hide.outliers`	boolean. Returns the outlier-free data if set to `TRUE`. Defaults to `FALSE`.
`lower`	an atomic vector. First element of `x` from `nleqslv`.
`upper`	an atomic vector. Second element of `x` from `nleqslv`.
`method.in`	See `method` in `nleqslv`
`global.in`	See `global` in `nleqslv`
`control.in`	See `control` in `nleqslv`

Details

The α-outlier region of a Weibull distribution is generally not available in closed form or via the tails, such that a non-linear equation system has to be solved.

Value

Data frame of the input data and an index named `is.outlier` that flags the outliers with `TRUE`. If hide.outliers is set to `TRUE`, a simple vector of the outlier-free data.

Author(s)

A. Rehage

References

Dodson, B. (2006) *The Weibull Analysis Handbook*. American Society for Quality, 2nd edition.

See Also

dweibull, nleqslv

Examples

```
# lifetime data example taken from Table 2.2, Dodson (2006)
temp <- c(12.5, 24.4, 58.2, 68.0, 69.1, 95.5, 96.6, 97.0,
          114.2, 123.2, 125.6, 152.7)
aout.weibull(temp, c(2.25, 97), 0.1)
```

---

citiesData — *Population of the 999 largest German cities*

Description

Population of the 999 largest German cities as a real life example for Pareto distributed data

Usage

```
data(citiesData)
```

Format

List with one element

References

http://bevoelkerungsstatistik.de

---

createDesMat — *Create design matrix for log-linear models of contingency tables*

Description

This function creates a design matrix for contingency tables and is particularly useful for log-linear Poisson models. It uses effect coding of the variables: First the rows of the contingency table from top to bottom, then the columns from left to right.

Usage

```
createDesMat(n, p)
```

Arguments

n	Number of rows of the corresponding contingency table.
p	Number of columns of the corresponding contingency table.

Value

A (n+p-1) times (n*p) design matrix.

Author(s)

A. Rehage

References

Kuhnt, S.; Rapallo, F.; Rehage, A. (2014) Outlier detection in contingency tables based on minimal patterns. *Statistics and Computing* 24 (3), 481-491.

Examples

```
createDesMat(3, 5)
```

`daysabs` *Number of absence days of students*

Description

Number of absence days of students

Usage

```
data(daysabs)
```

Format

Vector with 314 elements

References

http://www.ats.ucla.edu/stat/r/dae/nbreg.htm

Index